THREE DIMENSIONAL RECONSTRUCTION FROM OPTICAL FLOW USING TEMPORAL INTEGRATION

RAMESH RANGACHAR
Robotics Laboratory
Department of Mechanical Engineering
University of Maryland
College Park, Maryland 20742
Sensory Intelligence Group
and
Robot Systems Division

TSAI-HONG HONG
MARTIN HERMAN
Sensory Intelligence Group
Robot Systems Division

RANDALL LUCK
Aspex Incorporated
530 Broadway
New York, NY 10012

*U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
Manufacturing Engineering Laboratory
Gaithersburg, MD 20899

NIST

THREE DIMENSIONAL RECONSTRUCTION FROM OPTICAL FLOW USING TEMPORAL INTEGRATION

RAMESH RANGACHAR
Robotics Laboratory
Department of Mechanical Engineering
University of Maryland
College Park, Maryland 20742
Sensory Intelligence Group
and
Robot Systems Division

TSAI-HONG HONG
MARTIN HERMAN
Sensory Intelligence Group
Robot Systems Division

RANDALL LUCK
Aspex Incorporated
530 Broadway
New York, NY 10012

*U.S. DEPARTMENT OF COMMERCE
National Institute of Standards
and Technology
Manufacturing Engineering Laboratory
Gaithersburg, MD 20899

Three Dimensional Reconstruction from Optical Flow Using Temporal Integration

Ramesh Rangachar[1,3], Tsai-Hong Hong[1,2], Martin Herman[1], and Randall Luck[4]

[1]National Institute of Standards and Technology (NIST)
Bldg. 220, Rm B124, Gaithersburg, MD 20899

[2]Department of Computer Science and Information Systems
The American University, Washington, D.C. 20016

[3]Robotics Laboratory, Department of Mechanical Engineering
University of Maryland, College Park, MD 20742

[4]Aspex Incorporated, 530 Broadway, New York, NY 10012

ABSTRACT

Image flow, the apparent motion of brightness patterns on the image plane, can provide important visual information such as distance, shape, surface orientation, and boundaries. It can be determined by either feature tracking or spatio-temporal analysis. The optical flow thus determined can be used to reconstruct the 3-D scene by determining the depth from the camera of every point in the scene. However, the optical flow determined by either of the methods mentioned above will be noisy. As a result, it may be difficult to use the depth information obtained from optical flow in practical applications such as image segmentation, 3-D reconstruction, path planning, etc. By using temporal integration, we can increase the accuracy of both the optical flow and the depth determined from optical flow.

In this work, we describe an incremental integration scheme called the *running average method* to temporally integrate the image flow. We integrate the depth from camera obtained using optical flow determined from gradient based methods, and show that the results of temporal integration are much more useful in practical applications than the results from local edge operators. Finally, we consider an image segmentation example and show the advantages of temporal integration.

1. INTRODUCTION

Visual information such as distance, shape, orientation, and structure can be obtained using gradient based methods or feature based methods. Several algorithms have been proposed to determine the 3-D structure of the scene using local operators. The results obtained using local operators are noisy, and have a large error in individual measurements. Temporal integration is a method in which the individual values determined using local operators are integrated over a period of time to obtain consistent values for the quantity being determined.[4, 11] In section 2, we briefly describe two methods of determining the optical flow. In section 3, we describe the Epipolar Plane Image (EPI), and define terms used in subsequent discussions. We present the running average method of temporal integration in section 4. In section 5, we show how 3-D reconstruction, i.e., determination of depth from optical flow, is possible. Experiments and results follow (section 6), and we compare the results of temporal integration with the results from local edge operators. We also illustrate how image segmentation can be performed successfully using the results of temporal integration.

2. ESTIMATION OF OPTICAL FLOW

Image flow is defined as the apparent motion of brightness patterns on the retina of the eye (or on the image plane of a camera).[5] It can be determined using monocular vision (i.e., a single camera) using the following methods:

1. Feature matching based methods:

In these methods, identifiable features from a sequence of images are extracted and correspondence is established. The corresponding features are used to calculate a set of disparity vectors for the sequence. Any identifiable entity can be used as a feature, but sharp, localized features give the best accuracy.

2. Spatio-temporal methods.

These methods use the spatial and temporal relationships of image intensities to determine the optical flow. The time history of the position of a feature in the image can be represented in a 3-dimensional coordinate system (x,y,t). This is often called a spatio-temporal solid or spatio-temporal volume. The spatio-temporal area is defined as an $x-t$ slice of the solid, i.e., a slice taken with $y = constant$. The slope at any point (x,t) in the spatio-temporal area gives the horizontal component of its velocity. Therefore, the horizontal component of optical flow can be obtained by determining the orientation of the edges.

3. EPIPOLAR PLANE IMAGE

As discussed above, the time history of each feature point in the image gives the spatio-temporal volume, and a slice in the temporal direction gives the spatio-temporal area. Figure 1(a) shows a stationary scene containing a railroad car in the background and a tree in the foreground. Assume that the camera moves in the horizontal direction, that the optical axis of the camera points perpendicular to the direction of motion, and that the optical flow is in the negative direction. A horizontal scan line j of the image is selected and the spatio-temporal area for this scan line is obtained. When the scan line is coincident with the direction of motion of the camera, the spatio-temporal area is called the Epipolar Plane Image (EPI).[2] The EPI of the line j is shown in Figure 1(b). The slope corresponding to the tree (closer object) is smaller than the slope corresponding to the railroad car when the slope is measured from the time (vertical) axis. Thus, depending upon the depth of the objects along the scan line j, there will be lines of different slopes in the EPI. The greater the depth of the object from the camera, the greater the slope measured from the time axis will be.
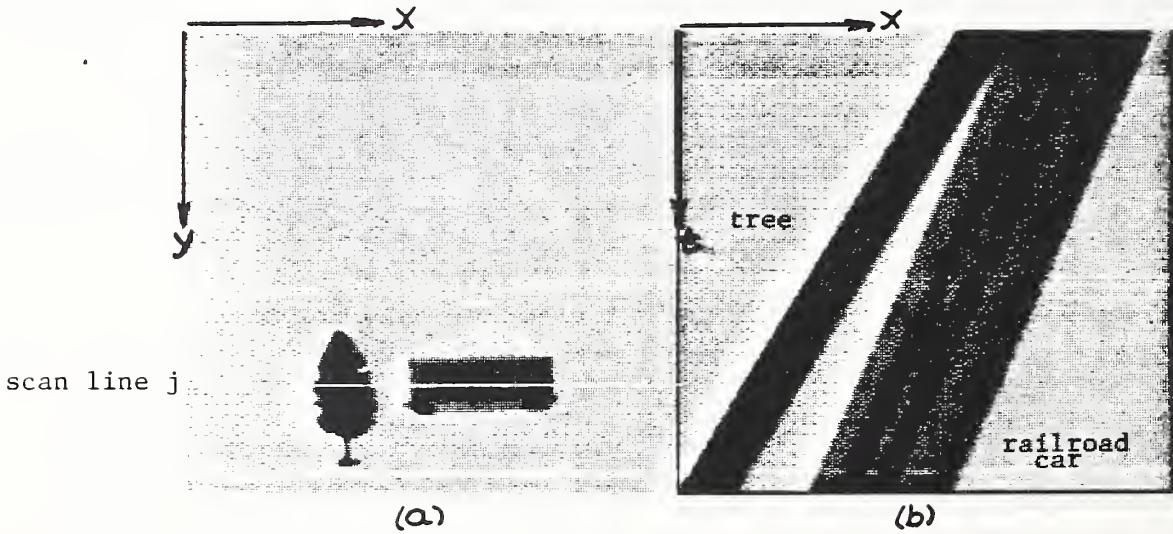

Figure 1: (a) Image at time k; (b) EPI for scan line j.

The slope of the edges, and hence the depth, can be measured using local edge operators.[6] An algorithm to estimate the depth from optical flow determined using local edge operators is presented in another paper by us in these proceedings.[13] Here, we use the results of local edge operators and explain how the results can be significantly improved using temporal integration. Throughout our discussion, we use EPIs such as the one shown in Figure 1(b), and show depth values at several space-time coordinates.

To familiarize the readers with the EPI and the terms we use in our discussion consider the sample segment shown in Figure 1(c), (also see Figure 1(b)). In the figure, the x-axis (towards right) represents space, and the t-axis (downward) represents time. In this example, the x and t axes are both labelled with values from 100 to 114. The unit of measure for the x axis is $pixels$. For the t axis, the unit is $frame-time$, with one frame-time being equal to $\frac{1}{30}$th of a second. Each horizontal line shows the intensity of the objects selected along the scan line at a particular instant

2

100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	
52	44	39	38	40	46	48	52	56	68	89	121	159	187	203	100
46	40	38	40	45	48	49	53	63	78	104	143	177	197	207	101
40	38	39	43	47	49	51	60	69	96	128	165	192	204	208	102
39	38	41	46	49	50	55	65	84	116	151	183	201	207	208	103
38	39	44	48	51	53	60	76	102	136	172	196	205	208	208	104
38	43	47	49	51	56	68	90	123	159	189	203	207	208	209	105
41	46	49	50	54	63	80	109	145	179	199	207	208	209	209	106
44	48	52	52	59	72	96	128	167	193	205	208	208	209	209	107
46	49	50	55	65	85	115	153	185	202	207	208	208	209	209	108
48	49	52	60	76	100	138	174	197	206	207	208	208	209	209	109
49	51	57	69	91	124	161	192	204	207	208	208	209	209	209	110
50	55	64	81	108	147	181	200	205	208	208	209	209	209	209	111
52	59	74	98	132	169	193	205	208	208	208	209	209	209	210	112
55	64	86	119	155	186	201	207	208	209	209	209	209	209	209	113
61	77	103	139	175	198	204	208	209	209	209	209	210	209	209	114

Fugure 1(c): Intensity values for a segment of Figure 1(b)

of time. For example, the first horizontal line shows the intensity values at time $t = 100$, for $x = 100$ to $x = 114$. Since the camera is moving, the intensity value at any x changes with respect to t. In Figure 1(c), as t increases, the objects in the scene, and hence the intensity values on the image plane, move to the left. For example, the intensity at $x = 106$ and $t = 108$ is 115. Let us write this as $I_{(106,108)} = 115$. Since the object is moving to the left (relative to the camera), at the next instant of time $t = 109$, it will have moved slightly to the left. In this example, let us say that it will move to $(105,109)$, with $I_{(105,109)} = 100$. These two points are called *corresponding points*.

Since the position of the camera is continuously changing, the *intensity* of corresponding points is not the same. However, since the camera is looking sideward and moving horizontally, the *depth* from the camera for all corresponding points should be the same. In the discussions that follow, we show that the depth results at corresponding points obtained using local edge operators have significant variations. The variations are so large that it may make the results unusable for some practical applications. We define temporal integration in the next section and show that it results in consistent values for optical flow computations.

4. TEMPORAL INTEGRATION

Temporal integration is the method of integrating corresponding local image value I obtained at consecutive time instants over a period of time to average the quantity being integrated. In this application, the quantity being integrated is the image flow u (which represents the depth d). To integrate two successive frames, we do the following.

1. Compute $u(x_i, t_j)$.
2. Use $u(x_i, t_j)$ to predict the position of the corresponding point (x_k, t_{j+1}) in the next frame in the EPI.
3. Compute $u(x_k, t_{j+1})$.
4. Translate the optical flow at (x_i, t_j) to (x_k, t_{j+1}). Thus, $u(x_i, t_j)$ and $u(x_k, t_{j+1})$ are available at (x_k, t_{j+1}) for further computation.
5. Determine the average of $u(x_i, t_j)$ and $u(x_k, t_{j+1})$. This is the integrated value at $u(x_k, t_{j+1})$.

To achieve real-time results, we use an incremental algorithm for temporal integration, known as *the running average method*. The algorithm for finding the running average is given below.

1. Determine the *optical flow*.
2. Create an image *count* according to the rule:

   FOR all pixels, IF *flow* > 0, *count* = 1 OTHERWISE, *count* = 0.
3. Initialize an *integrated optical flow* image and an *accumulated count* image.
4. Use the *integrated optical flow* to determine the correspondence with the *optical flow*.

3

5. Using the correspondence established, translate both the *integrated optical flow* and the *accumulated count*.

6. Obtain the *updated count* using

$$Updated\ count\ =\ count\ +Accumulated\ count$$

7. Calculate the integrated optical flow using

$$Integrated\ Optical\ Flow = \frac{Optical\ flow\ +Integrated\ Optical\ flow \times Accumulated\ count}{updated\ count}$$

Establishing the correspondence between two consecutive frames is a difficult problem. In this work, we have used optical flow values *less than* one pixel per frame-time. This is because larger optical flow values obtained using gradient based methods are known to induce large measurement errors due to undersampling.[7] Since the optical flow, or, *the apparent motion of brightness patterns on the image plane* is less than one pixel per frame-time, instead of determining the correspondence at every time instant, we update the ith, $(i+1)$th, and $(i-1)$th pixels at the next instant of time, only if a value for optical flow is present at this location.

5. DEPTH FROM OPTICAL FLOW

Consider a sideward-looking camera moving horizontally with a velocity V. In one time-instant, an object at P at a distance d from the camera moves to Q. At the same time, on the image plane, the image of the object moves from A to B, the distance being equal to the optical flow u.

If f is the focal length of the camera, using similar triangles OAB and OPQ, we get:

$$\frac{d}{f} = \frac{-V}{u}$$

$$u = \frac{-fV}{d}$$

If the camera is moving with a uniform velocity V, then the numerator of the above equation is a constant. The equation can be written as follows.

$$u = \frac{-K}{d} \qquad (1)$$
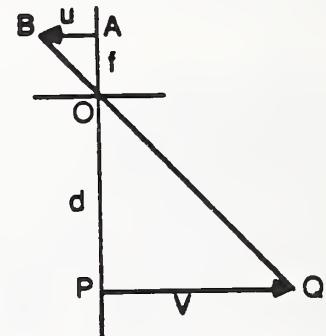
In the case of a horizontally moving camera where the motion is along the x axis, the component of motion in the y direction is zero. An equation for the optical flow u is given by:

$$u = \frac{-I_t}{I_x} \qquad (2)$$

Using Equations (1) and (2), an equation for the depth d of an object from the camera can be obtained.

$$d = \frac{KI_x}{I_t} \qquad (3)$$

In the discussions that follow, we have used Equation (3) to determine the depth of objects from the camera. For more information on the derivation of these equations, refer to the references listed in this paper.[13-15]

6. EXPERIMENTS, RESULTS AND DISCUSSION

A top view of the setup for our experiments is shown in Figure 2. We use an optical rail which has two platforms as shown in the figure: one translates horizontally and the other rotates about a vertical axis. The camera is mounted on the platforms as shown. The accuracy of the velocity of translation is within 0.02% of the selected velocity. Objects such as O_1 and O_2 can be placed in front of the camera. In our experiments, we placed a single object in front of the camera and obtained the EPI using a real-time, high-speed, image processing machine called Pipelined Image Processing Engine (PIPE). PIPE was conceived and designed at the National Institute of Standards and Technology (formerly National Bureau of Standards) by Kent, et al. [8] and is commercially available through Aspex Incorporated.[1] It was designed specifically for low level vision tasks at very high speed. It has 8 bit gray scale resolution and operates on 256×256 images at video rate (60 fields per second). It can also operate on images of larger size at lower rates. A

4
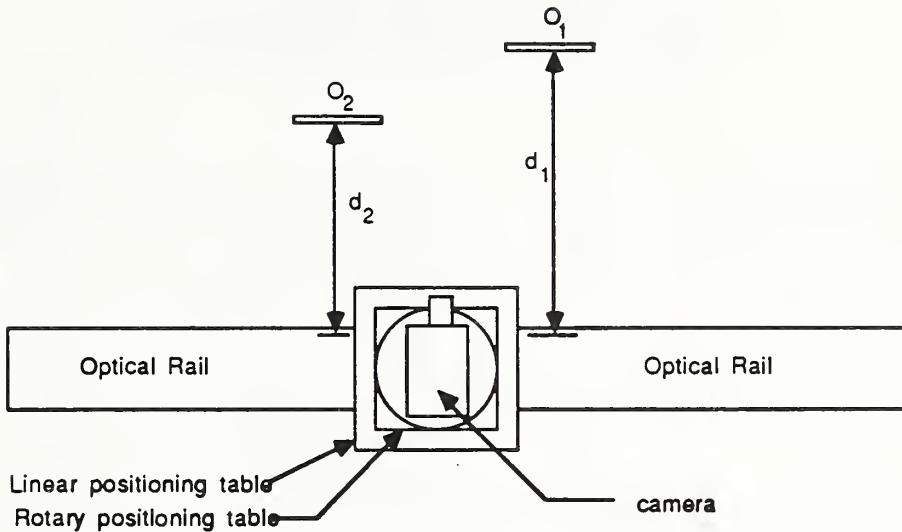
Figure 2: Experimental setup.

complete system can perform over one billion operations per second. For further details on PIPE, refer to the references listed in this paper.[3, 8-10, 12, 14, 16]

We have used PIPE to obtain the EPI at the rate of 30 fields per second. The EPI is used to determine the depth, and a program on the SUN computer is used to temporally integrate the results. The results of our experiments are given below.

Experiment #1:
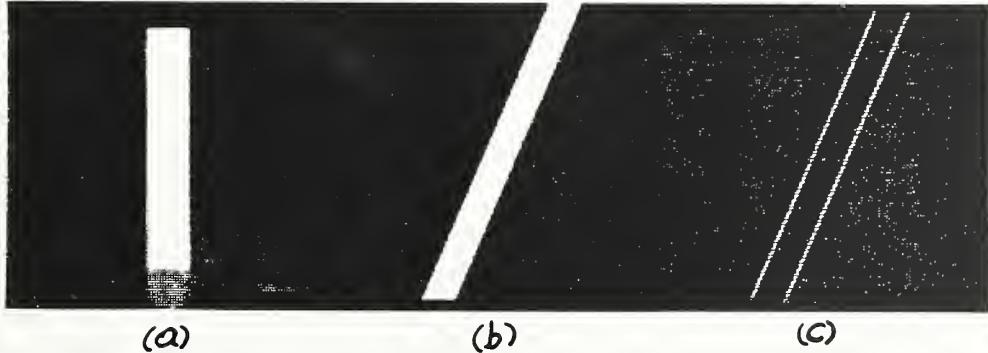
Figure 3: Experiment #1: (a) Original Scene; (b) EPI for (a); (c) optical flow for (b).

We placed an object in front of the camera as shown in Figure 3(a) and obtained its EPI (Figure 3(b)). We used the EPI to determine the depth using Equation (3) using a 5×5 local edge operator (see Figure 3(c)). The statistics for the whole image is given below. The mean represents the average distance (in millimeters) of the object from the camera determined using Equation (3). Figure 4 shows the depth values (in centimeters) for a small segment of the image.

5

Results Using Edge Operators		
Size	Mean (mm)	Standard Deviation (mm)
5×5	1239	104

150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	
0	0	0	0	0	0	0	0	0	0	131	0	0	0	0	18
0	0	0	0	0	0	0	0	0	138	120	0	0	0	0	19
0	0	0	0	0	0	0	0	0	131	0	0	0	0	0	20
0	0	0	0	0	0	0	0	0	122	0	0	0	0	0	21
0	0	0	0	0	0	0	0	134	110	0	0	0	0	0	22
0	0	0	0	0	0	0	0	128	0	0	0	0	0	0	23
0	0	0	0	0	0	0	137	118	0	0	0	0	0	0	24
0	0	0	0	0	0	0	130	0	0	0	0	0	0	0	25
0	0	0	0	0	0	0	119	0	0	0	0	0	0	0	26
0	0	0	0	0	0	130	103	0	0	0	0	0	0	0	27
0	0	0	0	0	0	122	0	0	0	0	0	0	0	0	28
0	0	0	0	0	132	110	0	0	0	0	0	0 ·	0	0	29
0	0	0	0	0	129	0	0	0	0	0	0	0	0	0	30
0	0	0	0	142	123	0	0	0	0	0	0	0	0	0	31
0	0	0	0	132	0	0	0	0	0	0	0	0	0	0	32

Fugure 4: Depth values for a segment of the EPI in Figure 3(b).

150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	
0	0	0	0	0	0	0	0	0	0	131	0	0	0	0	18
0	0	0	0	0	0	0	0	0	138	120	0	0	0	0	19
0	0	0	0	0	0	0	0	0	131	0	0	0	0	0	20
0	0	0	0	0	0	0	0	0	127	0	0	0	0	0	21
0	0	0	0	0	0	0	0	129	121	0	0	0	0	0	22
0	0	0	0	0	0	0	0	126	0	0	0	0	0	0	23
0	0	0	0	0	0	0	127	125	0	0	0	0	0	0	24
0	0	0	0	0	0	0	126	0	0	0	0	0	0	0	25
0	0	0	0	0	0	0	126	0	0	0	0	0	0	0	26
0	0	0	0	0	0	126	125	0	0	0	0	0	0	0	27
0	0	0	0	0	0	125	0	0	0	0	0	0	0	0	28
0	0	0	0	0	125	125	0	0	0	0	0	0	0	0	29
0	0	0	0	0	125	0	0	0	0	0	0	0	0	0	30
0	0	0	0	125	125	0	0	0	0	0	0	0	0	0	31
0	0	0	0	125	0	0	0	0	0	0	0	0	0	0	32

Figure 5: Integrated depth values for the EPI in Figure 3(b).

In the segment of the image shown in Figure 4, all the points are at the same distance from the camera. Therefore, we expect all points to have the same value for depth. However, in the results obtained using local operators, there is a large variation. In the small segment shown, the range for depth values is $(138 - 103) = 35cm$, or $350mm$. Because of this variation, the results may not be suitable for practical applications such as image segmentation, path planning, 3-D reconstruction, etc. Figure 5 shows the segment of the integrated image that corresponds to the segment shown in Figure 4. It shows the results of temporal integration on the depth values obtained from local edge operators. The integration was started at $t = 20$, and within 8 time-frames, the depth values become consistent. A graph of the average depth versus time is shown in Figure (6) for both segments. It shows how the results converge in the case of temporal integration, thus making it more suitable for practical applications.

Depth from Camera vs Frame-Time
(1 Frame-time=1/30 sec)
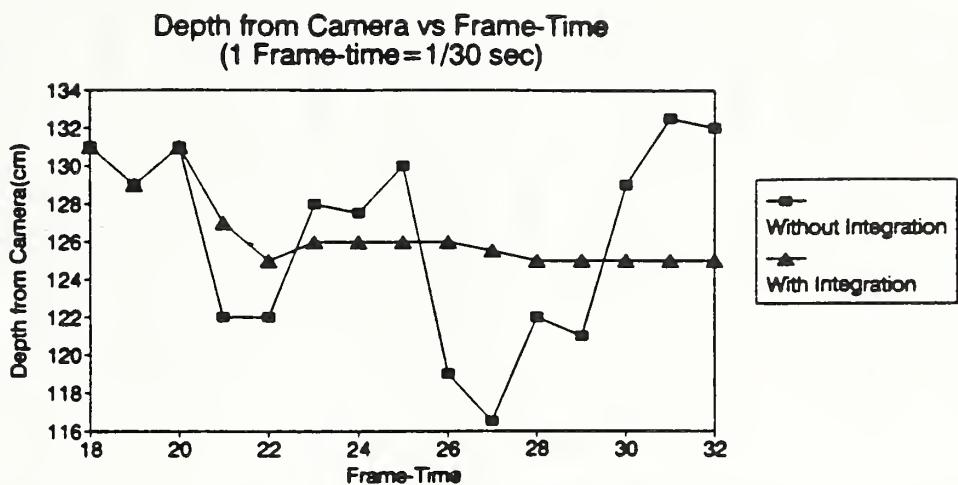
Figure 6: Graph of average depth vs time
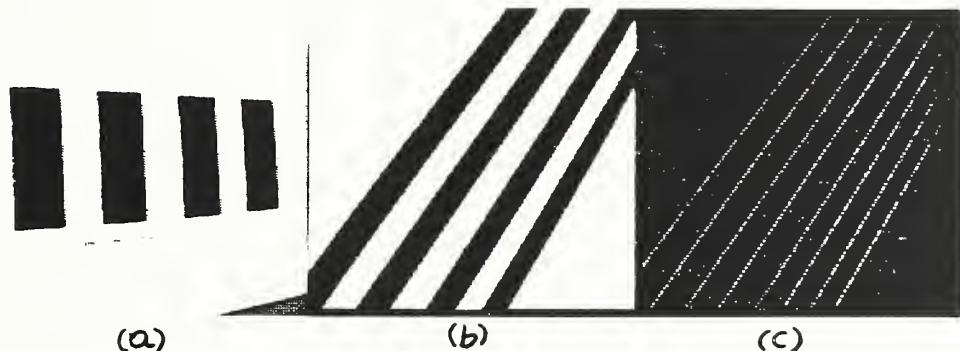
Figure 7: Experiment #2: (a) Original Scene; (b) EPI for (a); (c) optical flow for (b).

178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	
0	0	0	0	0	0	0	0	0	0	0	75	68	0	0	18
0	0	0	0	0	0	0	0	0	0	0	71	0	0	0	19
0	0	0	0	0	0	0	0	0	0	73	68	0	0	0	20
0	0	0	0	0	0	0	0	0	73	68	0	0	0	0	21
0	0	0	0	0	0	0	0	73	65	0	0	0	0	0	22
0	0	0	0	0	0	0	0	67	0	0	0	0	0	0	23
0	0	0	0	0	0	0	75	70	0	0	0	0	0	0	24
0	0	0	0	0	0	76	74	0	0	0	0	0	0	0	25
0	0	0	0	0	0	73	0	0	0	0	0	0	0	0	26
0	0	0	0	0	72	67	0	0	0	0	0	0	0	0	27
0	0	0	0	73	66	0	0	0	0	0	0	0	0	0	28
0	0	0	0	69	0	0	0	0	0	0	0	0	0	0	29
0	0	0	72	66	0	0	0	0	0	0	0	0	0	0	30
0	0	74	68	0	0	0	0	0	0	0	0	0	0	0	31
0	0	70	0	0	0	0	0	0	0	0	0	0	0	0	32

Figure 8a: Depth values for a segment of 7(b)

7

198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	
0	0	0	0	0	0	0	0	0	0	0	0	76	0	0	18
0	0	0	0	0	0	0	0	0	0	0	77	70	0	0	19
0	0	0	0	0	0	0	0	0	0	81	73	0	0	0	20
0	0	0	0	0	0	0	0	0	0	79	0	0	0	0	21
0	0	0	0	0	0	0	0	0	80	73	0	0	0	0	22
0	0	0	0	0	0	0	0	81	74	0	0	0	0	0	23
0	0	0	0	0	0	0	0	78	0	0	0	0	0	0	24
0	0	0	0	0	0	0	80	72	0	0	0	0	0	0	25
0	0	0	0	0	0	80	73	0	0	0	0	0	0	0	26
0	0	0	0	0	0	76	0	0	0	0	0	0	0	0	27
0	0	0	0	0	79	71	0	0	0	0	0	0	0	0	28
0	0	0	0	80	73	0	0	0	0	0	0	0	0	0	29
0	0	0	0	77	0	0	0	0	0	0	0	0	0	0	30
0	0	0	78	71	0	0	0	0	0	0	0	0	0	0	31
0	0	81	74	0	0	0	0	0	0	0	0	0	0	0	32

Fugure 8b: Depth values for a segment of 7(b).

221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	
0	0	0	0	0	0	0	0	0	0	0	84	0	0	0	18
0	0	0	0	0	0	0	0	0	0	79	76	0	0	0	19
0	0	0	0	0	0	0	0	0	91	83	0	0	0	0	20
0	0	0	0	0	0	0	0	0	86	0	0	0	0	0	21
0	0	0	0	0	0	0	0	84	78	0	0	0	0	0	22
0	0	0	0	0	0	0	0	80	0	0	0	0	0	0	23
0	0	0	0	0	0	0	86	79	0	0	0	0	0	0	24
0	0	0	0	0	0	87	81	0	0	0	0	0	0	0	25
0	0	0	0	0	0	79	0	0	0	0	0	0	0	0	26
0	0	0	0	0	80	72	0	0	0	0	0	0	0	0	27
0	0	0	0	0	78	0	0	0	0	0	0	0	0	0	28
0	0	0	0	83	78	0	0	0	0	0	0	0	0	0	29
0	0	0	84	80	0	0	0	0	0	0	0	0	0	0	30
0	0	0	80	0	0	0	0	0	0	0	0	0	0	0	31
0	0	82	74	0	0	0	0	0	0	0	0	0	0	0	32

Fugure 8c: Depth values for a segment of 7(b).

178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	
0	0	0	0	0	0	0	0	0	0	0	75	68	0	0	18
0	0	0	0	0	0	0	0	0	0	0	71	0	0	0	19
0	0	0	0	0	0	0	0	0	0	73	68	0	0	0	20
0	0	0	0	0	0	0	0	0	73	70	0	0	0	0	21
0	0	0	0	0	0	0	0	73	70	0	0	0	0	0	22
0	0	0	0	0	0	0	0	71	0	0	0	0	0	0	23
0	0	0	0	0	0	0	71	71	0	0	0	0	0	0	24
0	0	0	0	0	0	71	71	0	0	0	0	0	0	0	25
0	0	0	0	0	0	71	0	0	0	0	0	0	0	0	26
0	0	0	0	0	71	71	0	0	0	0	0	0	0	0	27
0	0	0	0	71	71	0	0	0	0	0	0	0	0	0	28
0	0	0	0	71	0	0	0	0	0	0	0	0	0	0	29
0	0	0	71	71	0	0	0	0	0	0	0	0	0	0	30
0	0	71	71	0	0	0	0	0	0	0	0	0	0	0	31
0	0	71	0	0	0	0	0	0	0	0	0	0	0	0	32

Fugure 9a: Integrated depth values corresponding to 8a.

198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	
0	0	0	0	0	0	0	0	0	0	0	0	76	0	0	18
0	0	0	0	0	0	0	0	0	0	0	77	70	0	0	19
0	0	0	0	0	0	0	0	0	0	81	73	0	0	0	20
0	0	0	0	0	0	0	0	0	0	78	0	0	0	0	21
0	0	0	0	0	0	0	0	0	79	77	0	0	0	0	22
0	0	0	0	0	0	0	0	79	77	0	0	0	0	0	23
0	0	0	0	0	0	0	0	78	0	0	0	0	0	0	24
0	0	0	0	0	0	0	78	78	0	0	0	0	0	0	25
0	0	0	0	0	0	78	78	0	0	0	0	0	0	0	26
0	0	0	0	0	0	78	0	0	0	0	0	0	0	0	27
0	0	0	0	0	78	78	0	0	0	0	0	0	0	0	28
0	0	0	0	78	78	0	0	0	0	0	0	0	0	0	29
0	0	0	0	78	0	0	0	0	0	0	0	0	0	0	30
0	0	0	78	78	0	0	0	0	0	0	0	0	0	0	31
0	0	78	78	0	0	0	0	0	0	0	0	0	0	0	32

Fugure 9b: Integrated depth values corresponding to 8b.

221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	
0	0	0	0	0	0	0	0	0	0	0	84	0	0	0	18
0	0	0	0	0	0	0	0	0	0	79	76	0	0	0	19
0	0	0	0	0	0	0	0	0	91	83	0	0	0	0	20
0	0	0	0	0	0	0	0	0	87	0	0	0	0	0	21
0	0	0	0	0	0	0	0	86	85	0	0	0	0	0	22
0	0	0	0	0	0	0	0	0	85	0	0	0	0	0	23
0	0	0	0	0	0	0	85	84	0	0	0	0	0	0	24
0	0	0	0	0	0	85	84	0	0	0	0	0	0	0	25
0	0	0	0	0	0	84	0	0	0	0	0	0	0	0	26
0	0	0	0	0	84	84	0	0	0	0	0	0	0	0	27
0	0	0	0	0	84	0	0	0	0	0	0	0	0	0	28
0	0	0	0	84	84	0	0	0	0	0	0	0	0	0	29
0	0	0	84	84	0	0	0	0	0	0	0	0	0	0	30
0	0	0	84	0	0	0	0	0	0	0	0	0	0	0	31
0	0	84	84	0	0	0	0	0	0	0	0	0	0	0	32

Fugure 9c: Integrated depth values corresponding to 8c.

Experiment #2:

This experiment shows that the results of temporal integration can be used successfully for practical applications such as image segmentation. Figure 7(a) shows the object, which is an inclined white board containing a number of black stripes. The EPI for this image is shown in Figure 7(b). It can be seen from the EPI that the stripes have different slopes because they are at different distances from the camera. The depth from camera is shown in 7(c). Figures 8a-8c show three segments of the depth values from 7(c). It can be seen that it may be difficult to do segmentation of the stripes based on the depth values from simple local edge operators.

Figures 9a-9c show integrated results corresponding to 8a-c respectively. It can be seen that the results of temporal integration converge within 6-8 frames, and thus can be used to segment the image 7(a) using simple thresholding. The integrated output can also be used in other practical applications.

7. CONCLUSION

Image flow can be used for reconstruction of 3D scenes, but the results obtained using local operators will be very noisy and inaccurate. A scheme to integrate the results over a period of time will yield better results. Because of the volume of data involved in image processing, incremental integration methods used for temporal integration offer real-time performance. The results from temporal integration, when compared to those obtained without integration, show that considerable accuracy can be obtained using temporal integration. The individual depth values from

9

segments of EPI show that the output of temporal integration becomes consistent after integration over 6-8 frames. Experiments performed in our laboratory have shown that the integrated values are consistent and can be successfully used for image segmentation.

ACKNOWLEDGEMENTS

REFERENCES

1. *The PIPE User's Manual*, Aspex, 1987. 530 Broadway, New York, NY 10012

2. Bolles, R.C., Baker, H.H., and Marimont, D.H., "Epipolar-Plane Image Analysis: An Approach to Determining Structure from Motion," *International Journal of Computer Vision 1:7-55*, 1987.

3. Herman, M., "Application of the PIPE Image Processing Machine to Scanning Microscopy," *Proc. SPIE Scanning Microscopy Technologies and Applications*, vol. 897, pp. 169-173, Los Angeles, CA., January 1988.

4. Albus, J.S. and Hong, T.-H., "Motion, Depth and Image Flow," *IEEE Robotics & Automation Conference*, May 13-18, 1990.

5. Horn, B.K.P. and Schunk, B.G., "Determining Optical Flow," *Artificial Intelligence*, vol. 17, pp. 185-203, 1981.

6. Jain, A. K., *Fundamentals of Digital Image Processing*, Prentice-Hall, Inc., Englewood Cliffs, New Jersy, 1989.

7. Kearney, J.K., Thompson, W.B., and Boley, D.L., "Optical Flow Estimation: An Error Analysis of Gradient-Based Methods with Local Optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 2, pp. 229-244, March 1987.

8. Kent, E.W., Shneier, M.O., and Lumia, R., "PIPE (Pipelined Image Processing Engine)," *J. Parallel and Distributed Computing*, pp. 50-78, 1985.

9. Luck, R.L., "PIPE: a Parallel Processor for Dynamic Image Processing," in *Image Understanding and Man-Machine Interface*, ed. Pearson, J.J. Barett, E., pp. 109-115 , Proc. SPIE 758, 1987.

10. Luck, R.L., "An overview of the PIPE Systems," *Proc. Third Intl. Conference on Super Computing*, vol. 3, pp. 69-78, Boston, 1988.

11. Matthies, L.H., Szeliski, R., and Kanade, T., "Kalman Filter Based Algorithms for Estimating Depth from Image Sequences," *Technical Report, CMU-CS-87-195*, Computer Science Department, Carnegie Mellon University, Dec. 1987.

12. Nashman, M. and Chaconas, K., *Low Level Image Processing Techniques Using the Pipelined Image Processing Engine in the Flight Telerobotic Servicer*, Robot Systems Division, National Institute of Standards and Technology, 1989.

13. Rangachar, R., Hong, T.-H., Herman, M., and Luck, R.L., "Analysis of Optical Flow Estimation Using Epipolar Plane Images," *Proc. SPIE Symposium on Laser Science and Optics Applications*, Boston, Nov. 4-9, 1990.

14. Rangachar, R., Hong, T.-H., Herman, M., and Lupo, J., "Real-Time Implementation of a Differential Range Finder," *Proc. SPIE Real-Time Image Processing II: Algorithms, Architectures, and Applications*, Orlando, Florida, April 1990.

15. Rangachar, R., Hong, T.-H., Herman, M., Luck, R.L., and Lupo, J., "Real-Time Differential Range Estimation Based on Time-Space Imagery Using PIPE," *Proc. SPIE Real-Time Image Processing II: Algorithms, Architectures, and Applications*, Orlando, Florida, April 1990.

16. Singh, A., "Image Processing on PIPE," Technical Report TN-87-093, Philips Laboratories, Briarcliff Manor, New York, 1987.

4. TITLE AND SUBTITLE

Three Dimensional Reconstruction from Optical Flow Using Temporal Integration

5. AUTHOR(S)

Ramesh Rangachar, Tsai-Hong Hong, Martin Herman and Randall Luck

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

Image flow, the apparent motion of brightness patterns on the image plane, can provide important visual information such as distance, shape, surface orientation, and boundaries. It can be determined by either feature tracking or spatio-temporal analysis. The optical flow thus determined can be used to reconstruct the 3-D scene by determining the depth from camera of every point in the scene. However, the optical flow determined by either of the methods mentioned above will be noisy. As a result, the depth information obtained from optical flow can not be successfully used in practical applications such as image segmentation, 3-D reconstruction, path planning, etc. By using temporal integration, we can increase the accuracy of both the optical flow and the depth determined from optical flow.

In this work, we describe an incremental integration scheme called the running average method to temporally integrate the image flow. We integrate the depth from camera obtained using optical flow determined from gradient based methods, and show that the results of temporal integration are much more useful in practical applications than the results from local edge operators. Finally, we consider an image segmentation example and show the advantages of temporal integration.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

3-D Reconstruction, Depth from camera, EPI, Feature matching, Gradient based method, Optical Flow, PIPE, Segmentation, Spatio-temporal, Temporal Integration.